Remarks

The present amendment responds to the Official Action dated April 13, 2005. The Official Action rejected claims 34 and 53 based on the judicially created doctrine of obviousness-type double patenting over claims 1 and 9 of U.S. Patent No. 6,366,999. Claims 34-40, 53, and 56-61 were rejected under 35 U.S.C. §103(a) based on Dibrino et al. U.S. Patent No. 6,061,707 (Dibrino) in view of Pawate et al. U.S. Patent No. 5,528,550 (Pawate). Claims 62, 64, and 65 were rejected under 35 U.S.C. §102(e) as being anticipated by Dibrino. Claims 63 and 66 were rejected under 35 U.S.C. §103(a) based on Dibrino in view of Pawate. These grounds of rejection are addressed below. Furthermore, since the rejection of claims 63 and 66 is the same rejection used in claims 62, 64, and 65, these two rejections will be combined and addressed below as one rejection.

A terminal disclaimer to obviate the provisional double patenting rejection is being filed with this response. Authorization to charge Deposit Account No. 50-1058 for the terminal disclaimer fee of $130 under 37 C.F.R. §1.20(d) accompanies this response. Claims 1-33 and 41-52 have been previously cancelled. Claims 34-40 and 53-66 are presently pending.

The Art Rejections

As addressed in greater detail below, Dibrino and Pawate do not support the Official Action's reading of them and the rejections based thereupon should be reconsidered and withdrawn. Further, the Applicants do not acquiesce in the analysis of Dibrino and Pawate

made by the Official Action and respectfully traverse the Official Action's analysis underlying its rejections.

Claims 34-40, 53, 56-61, 63, and 66 were rejected under 35 U.S.C. §103(a) based on Dibrino in view of Pawate. Dibrino's reference date is January 16, 1998 while the present invention's priority date is January 28, 1998. It is not admitted that Dibrino is in fact prior art. Nonetheless, the present response addresses fundamental differences between the present invention and Dibrino rather than undertaking the burdens of swearing behind Dibrino.

Claim 34 recites "instruction control lines derived from a registered instruction in a processor pipeline, the instruction control lines including conditional execution control lines to control conditional operation as specified in an instruction." The conditional execution control lines advantageously allow a programmer to specify through condition opcodes, as specified in an instruction a request, to generate an arithmetic condition flag (ACF) on a resulting condition which may occur from the execution of the instruction. With this approach, instructions may conditionally execute based on the state of an ACF or a linear combination of ACFs. See Specification, page 3, lines 28- 34 and Fig. 5A. The ACFs, unlike scalar flags such as carry, overflow, sign, and zero, are derived as specified in an instruction from the side effects of execution. Specification, page 9, line 21 – page 10, line 2. Claim 34 further recites "a second latch connected to the conditional execution control lines for holding instruction control signals for the instruction after the instruction has finished its execution state." Claim 34 further recites "an arithmetic condition flag (ACF) generation unit for providing a Boolean combination of a

present selected state with a previous state." The relied upon art does not teach and does not make obvious conditional execution as claimed.

Dibrino addresses an apparatus for generating an end-around carry in a floating point pipeline within a computer system. To this end, Dibrino's apparatus includes a shift-comparison logic circuit, a sign-comparison circuit, and a logic gate. The shift-comparison logic circuit produces a shift-count signal, and the sign-comparison logic circuit produces an operation signal. Coupled to the shift-comparison logic circuit and the sign-comparison logic circuit, the logic gate combines the shift-count signal and the operation signal with a carry-out signal generated by an end-around carry adder to provide an end-around carry signal for the end-around carry adder. Dibrino, col. 1, line 65 – col. 2, line 6 and Fig. 2. Dibrino's focus is entirely different than the presently claimed invention, and does not provide "instructional control lines including conditional execution control lines to control conditional operation as specified in an instruction," as claimed by claim 34, for example.

The Official Action relies on Dibrino at col. 3, lines 17-29, the operand portions 11 and 12 and the shift compare logic 28 in Fig. 2, and the shift count in Fig. 3 as purportedly teaching the conditional execution feature as claimed. Applicants respectfully disagree. Dibrino does not teach and does not suggest "instruction control lines derived from a registered instruction in a processor pipeline, the instruction control lines including conditional execution control lines to control conditional operation as specified in an instruction," as claimed in claims 34 and 53. See also claim 62 which recites "conditional execution control signals derived from an instruction in an instruction pipeline to control conditional operation as specified in the instruction." Even if,

for the sake of argument, an end around carry bit as taught in Dibrino is analogous to accomplishing conditional execution through an arithmetic condition flag generation unit as the Official Action suggests, Dibrino does not teach and does not suggest an instruction specifying the control operation for conditional execution as claimed. Furthermore, such an end around carry bit does not constitute a flag which is settable by an instruction.

Pawate fails to cure the deficiencies of Dibrino as a reference. Pawate addresses an apparatus for implementing a memory embedded search arithmetic logic unit. To this end, Pawate's apparatus includes control circuitry operable in response to control instructions received from a broadcast memory which controls the transfer of a word of data from data memory to a search circuitry. Pawate, col. 1, line 66 – col. 2, line 3. The search circuitry is operable to test the word of data retrieved from the data memory by applying a branch condition selected from a set of branch conditions defined by search instructions set held in the broadcast memory. Pawate, col. 2, lines 5-8. Pawate's apparatus addresses a completely different problem than accomplishing conditional execution as claimed.

The Official Action cites the direct feedback path from the accumulator of Fig. 3 and col. 6, lines 16-34 of Pawate as purportedly disclosing "an ACF latch for storing the previous state and feeding the previous state back to the ACF generation unit," as claimed. Applicants respectfully disagree. Pawate describes nothing of the sort. Pawate does not teach and does not disclose ACFs as disclosed in the present specification and described above. Pawate merely discloses an ALU 60 which applies one or more tests, such as those listed in TABLE I, to the result of the arithmetic operation on data from random access memory and accumulator 62.

Pawate, col. 6, lines 20-24. Pawate's disclosure does not address the problem of conditional execution control, nor does Pawate's disclosure have a reason to do so. Thus, Pawate's disclosure does not teach and does not disclose "an ACF latch for storing the previous state and feeding the previous state back to the ACF generation unit," as claimed.

Even if Pawate is combined with Dibrino as the Official Action suggests, the combination fails to meet the terms of the claimed features. Dibrino and Pawate, taken separately or in combination, do not teach and do not suggest "instruction control lines derived from a registered instruction in a processor pipeline, the instruction control lines including conditional execution control lines to control conditional operation as specified in an instruction," as claimed. Dibrino and Pawate, taken separately or in combination, do not teach and do not suggest "an arithmetic condition flag (ACF) generation unit for providing a Boolean combination of a present selected state with a previous state; and an ACF latch for storing the previous state and feeding the previous state back to the ACF generation unit," as claimed in claim 34.

Claims 62, 64, and 65 were rejected under 35 U.S.C. §102(e) as being anticipated by Dibrino. The Official Action relies on Fig. 3 and col. 3, lines 17-29 of Dibrino as purportedly disclosing "conditional execution control signals derived from an instruction in an instruction pipeline to control conditional operation as specified in the instruction" as claimed in claim 62. Applicants respectfully disagree. The cited portion of text merely addresses utilizing a 26-bit incrementer rather than a 48-bit end-around carry (EOC) when performing a floating point operation. Furthermore, Fig. 3 of Dibrino merely addresses providing a shift comparison logic to determine whether a B-operand lies entirely "underneath" an A*C operand. When shifting is

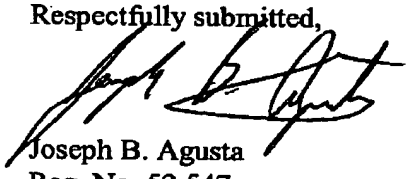
determined, it is used to effect floating point operation. Quite simply, Dibrino's floating point apparatus does not address the problem of conditional execution. Thus, Dibrino does not teach and does not make obvious "conditional execution control signals derived from an instruction in an instruction pipeline to control conditional operation as specified in the instruction," as claimed in claim 62.

The relied upon references fail to recognize and address the problem of conditional execution in the manner advantageously addressed by the present claims. The present claims are not taught, are not inherent, and are not obvious in light of the art relied upon.

Conclusion

All of the presently pending claims, as amended, appearing to define over the applied references, withdrawal of the present rejection and prompt allowance are requested.

Respectfully submitted,



Joseph B. Agusta
Reg. No. 52,547
Priest & Goldstein, PLLC
5015 Southpark Drive, Suite 230
Durham, NC 27713-7736
(919) 806-1600